Gantry Patient Data Base (GaPaDaBa) – concept and realization of an integrated, process oriented, data base driven information system for proton therapy

F. Emert[1], R. Besson[2], A. Lomax[1], A. Coray[1], D. Gruntz[2]

[1]Center of Proton Radiation Therapy, Paul Scherrer Institute, Villigen, Switzerland

[2]IMDS, University of Applied Science NWCH, Windisch, Switzerland

INTRODUCTION:

At PSI the GaPaDaBa project was launched in January 2005 at the Center of Proton Radiation Therapy (CPT). After treating 262 tumor patients until the end of 2005 with the existing gantry, the new PROSCAN project will increase the treatment capacity by a factor of 3 to 4 to about 150-200 patients per year by 2008/2009. This rise provides new challenges for the administrative, logistic, operational, and scientific processes within the center. Especially because of our extremely heterogeneous hard- and software infrastructure the resulting process-related complexity and intricate IT topology complicates, or even prevents, the use of conventional medical information systems.

After extensively dealing with commercial software solutions we now propose an evolutionary, self-made concept of an information system that

1. supports the high level of human interaction in the patient's course through a proton treatment,
2. supports the idea of total transparency of all processes,
3. combines sophisticated IT tools and methods with proton therapy specific requirements, and
4. follows the approach of open source software.

The feasibility of the concept was first tested on dummy patients and has recently become an integral part of our daily clinical routine.

Finally it is the ultimate goal of GaPaDaBa to develop a non-invasive, virtual control system in combination with an integrated, DB based information platform that allows

- to dynamically track, plan, and schedule the patient's route through the complex processes of a proton treatment,
- to link data from different sources (administration, treatment, operation, planning, monitoring),
- to solve internal (related to their origin) and overall queries,
- to support the analysis and the course of scientific, operational, and QA processes proactively.

METHODS AND MATERIAL:

Workflow

In general we consider a workflow to be the time dependent movement of documents, data, and/or tasks through a work process that can be modeled and analyzed using graph-based formalisms. Focusing on the operational aspect, the workflow characteristics can be narrowed down by answering the following questions:

1. How are tasks structured?
2. Who performs them?
3. What is their relative order?
4. How are they synchronized?
5. How does information flow to support the tasks?
6. How are tasks tracked?

To overcome underlying modeling limitations we merged aspects of the UML standard and the concept of event-driven process chains to a new dialect called GaPaDaBaMoLa (GaPaDaBa Modeling Language) integrating both multiple-role-based activities and event-driven logic.

In order to consequently follow a process-oriented approach in medical information systems, one has to apply the rules and regularities of the so called *workflow management cycle* (WFMC). The WFMC contains a series of measures having to be passed either sequentially or iteratively, starting with the following order: analysis – modeling – optimization – design – management – control.

State event machine

Generally a state event machine is any device that stores the status of something at a given time and can operate on inputs that change the status, and/or cause an action or output to take place for any given change. A more specific type is the non-deterministic finite state machine which has the additional features of a multiple start state and condition-dependent transitions into one or more states.

Therefore the GaPaDaBa state event machine can be described as a non-deterministic finite state machine that is event-driven. State transitions are triggered by events not by specific inputs. The GaPaDaBa state event machine serves as an interactive pattern for simulating the patient workflow. This approach of modeling and projecting the entire patient workflow, which is scaled down to the granularity of elementary units of work, onto an event-driven non-deterministic finite state machine represents one of the most important design features of GaPaDaBa.

Application architecture

The main application architecture is designed as a common 3-tier structure (Table 1), containing an Oracle database with hibernate (object-relational mapping) as data layer, the widely used, lightweight J2EE framework Spring as middle tier and a web front end as user interface layer. The application is running in a Tomcat application server on a Linux operating system. The whole application is separated in many stand-alone web applications, each representing a logical unit. They are merged together in a portal (see GUI). The applications follow the paradigm of a service oriented architecture and are accessible via web services from any other web application and also from rich client applications.

Layer	Contents
Presentation Layer	Data Presentation
	Data Input
	User Control
Business Logic Layer	Data Processing
	User Command Processing
Data Layer	Data Loading
	Data Persistence

Table 1: 3-tier-architecture

Graphical user interface

The web-based GaPaDaBa-GUI integrates the functionality of an interactive patient tracking tool and an electronic patient folder. The so called Patient Tracker serves as a monitor and instrument to initiate, propagate, and supervise the patient's progress through the workflow, i.e. the state event machine respectively. Since we follow a process oriented approach, the data collection for the electronic patient folder goes along with the patient's propagation through the workflow. The GUI in general functions as a portal and is developed along commonly accepted guidelines for an intuitive look-and-feel of GaPaDaBa.

Data scheme

The GaPaDaBa data scheme reflects the use case categories *administration*, *analysis*, *treatment*, *operation*, *planning*, *monitoring*, and *scheduling*. Due to the fact that GaPaDaBa follows an iterative model approach the data model is extended step by step taking into account the necessity of annoying data migrations between different software releases. Since the workflow potentially changes, the design of the data scheme has to guarantee both flexibility and robustness related to the appearing modifications. In order to achieve the goal of combining patient treatment related information with technical QA data it is necessary to follow a top-down design strategy with the patient positioned in the center of the data structure.

RESULTS:

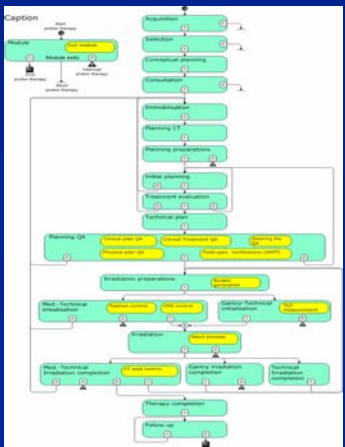

Figure 1: Unitized patient workflow

Starting the WFMC with the analysis phase we were forced to overcome inherent limitations of the UML standard and the modeling concept based on event-driven process chains. Due to their restricted capabilities of integrating multi-role-based activities in a single model view we defined GaPaDaBaMoLa, the GaPaDaBa modeling language. Describing the entire work- and dataflow using GaPaDaBaMoLa provides approximately 100 letter-sized pages of output structured in about 35 product functions. In order to scale down this amount of structural information we unitized the workflow as given in Figure 1.

Not visible here are the different roles we identified: *physician*, *medical physicist*, *physicist on-duty*, *therapy planner*, *IT agent*, *secretary*, *radiographer*, *quality manager*, *documentalist*.

The displayed scalable representation of the logical sequence of larger process units within the patient workflow initiated the idea of using a state event machine to flexibly implement the entire system based on an XML configuration. The currently implemented state event machine contains about 40 states with corresponding rules and conditions. The machine is event-driven and role-based, i.e. each personal account is related to one or more specified roles with predefined permissions depending on the state.

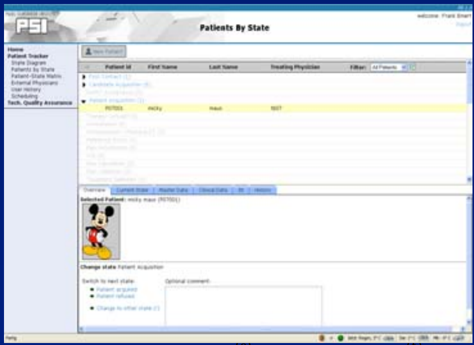

Figure 2: Patient Tracker mode (left: navigation menu; upper half: distribution of patients over list of states; lower half: electronic patient folder)

Figure 2 displays the portal view of the GUI in 'Patient Tracker' mode. Its sub items provide access to

1. the state diagram,
2. the dynamic patients-by-state-overview and tracking facilities including the electronic patient folder,
3. the overall patient progress (Figure 3; patient state matrix),
4. the data of external physicians,
5. the user history (protocol list of performed actions), and
6. a view onto the scheduling activities in the center which are part of a separate application but share the same database.

Figure 3: Patient state matrix (patient progression board)

Besides the entry page ('Home') we have currently implemented the 'Technical Quality Assurance' mode which contains detailed overviews, data, and information of the regularly performed daily, weekly, monthly, and yearly QA checks of the gantry and the beam line. In this context GaPaDaBa will serve as a QA documentation and reference tool.

Since the major effort of the last two years went into the concept, development and implementation of the entire GaPaDaBa framework "only" about 40% of the modules shown in Figure 1 have been implemented up to now. Nevertheless the realization time for new modules decreases significantly. The completion of the GaPaDaBa workflow and database implementation within the PRO-SCAN project is scheduled for end of 2008.

DISCUSSION:

As indicated GaPaDaBa aims to be a non-invasive, virtual control system in combination with an integrated, DB based information platform that provides total transparency for all patient related, technical and QA processes including their corresponding data sets. In its current state of development GaPaDaBa combines a state event machine and its underlying database with a sophisticated, web based graphical user interface that allows interactive workflow control of the patient as well as data collection and manipulation in an electronic patient folder.

To our knowledge this approach is unique in the field of proton radiation therapy. In tests with dummy patients tracked through the entire patient workflow and since starting patient treatment again in 2007 the current implementation of GaPaDaBa has shown high system availability (DB, application server), a precise tracking functionality, an almost trouble-free data input/output and high tolerance against inaccurate GUI handling. This system stability in combination with the good acceptance of GaPaDaBa in our team encourages us to continue and extend the development to our OPTIS-2 project (proton treatment of ocular tumors). Furthermore we believe GaPaDaBa to provide a methodological set of measures that has proven to be feasible and that in principle can be adapted to other proton sites, especially because it is highly based on open source components.